One-shot field spotting on colored forms using subgraph isomorphism

Maroua Hammami, Pierre Héroux, Sébastien Adam, Vincent Poulain d'Andecy

▶ **To cite this version:**

Maroua Hammami, Pierre Héroux, Sébastien Adam, Vincent Poulain d'Andecy. One-shot field spotting on colored forms using subgraph isomorphism. International Conference on Document Analysis and Recognition, 2015, Nancy, France. pp.586-590, 10.1109/ICDAR.2015.7333829 . hal-01249470

HAL Id: hal-01249470
https://hal.science/hal-01249470

Submitted on 1 Jan 2016

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

One-Shot field spotting on colored forms using subgraph isomorphism

Maroua Hammami – Pierre Héroux – Sébastien Adam
University of Rouen – LITIS EA 4108
Avenue de l'Université, 76801 Saint-Etienne-du Rouvray, France
Email: {maroua.hammami,pierre.heroux,sebastien.adam}@litislab.eu

Vincent Poulain d'Andecy
Itesoft
Parc d'Andron Le Séquoia 30470 Aimargues, France
Email: Vincent.PoulaindAndecy@itesoft.com

***Abstract*—This paper presents an approach for spotting textual fields in commercial and administrative colored forms. We proceed by locating these fields thanks to their neighboring context which is modeled with a structural representation. First, informative zones are extracted. Second, forms are represented by graphs. In these graphs, nodes represent colored rectangular shapes while edges represent neighboring relations. Finally, the neighboring context of the queried region of interest is modeled as a graph. Subgraph isomorphism is applied in order to locate this ROI in the structural representation of a whole document. Evaluated on a 130-document image dataset, experimental results show up that our approach is efficient and that the requested information is found even if its position is changed.**

I. Introduction

Automatic reading of commercial and administrative documents such as forms or invoices allows organisations to feed their information systems without time comsuming and tedious human interventions which may also introduce input errors. It consists in locating, extracting and recognizing targetted information which are added on an originally empty document (the form background).

To fulfill these objectives, a reading model is needed. It has to define the location of the information to be read and it is sometimes extended with the meaning of this information (Family name, subscriber id, address, zip code, phone number, amount, date...) in order to guide the recognition step or verify its results.

The location part of the model is of great importance since information location strongly conditions the subsequent processing. However, its definition is complicated by the variability in the way the data to be read appears on document. Indeed, even for several instances of the same form, the position of the information can vary according to the digitization conditions. It is particularly true in the context of "on demand" recognition where documents are digitized by distributed customers and recognized by a centralized reading system. Moreover, the information to be read is frequently handwritten. Thus, ink colors, thicknesses of the pen trace, size and syntax of the field to be read cannot be perfectly controled (i.e. several syntaxes can be used to express the same date, an amount field may contain a different number of digits on different instances of the same form). All these constraints make impossible the use of a fixed spatial template to describe the reading model.

In this article, we propose a relative reading model and the associated flexible algorithm to robustly locate informative zones on a target document. The approach copes with the requirements of commercial applications which impose that (i) the size and position may vary between several instances of the same form and (ii) the reading model is inputed through a user friendly human interface and is reduced to a single ground-truthed example.

In this context, the proposed model express the neighboring context of informative zones based on surrounding color rectangles. Indeed, as they are printed on the form background, color rectangles and frames are part of the document structure and can then be used as references for relative location. The neighboring context is then modeled as a graph whose nodes represent rectangles located around the queried information. In this graph, edges represent some visibility relations. A similar model is used to represent the structure of the whole target document on which the information is looked up. Then, the spotting procedure turns into an error-tolerant subgraph isomorphism problem (*i.e.* find the subgraph of the whole document representation that matches the best with the graph representing the context of the searched information).

The remaining of the paper is organized as follows. Section II presents the framework in which the contribution presented in this article takes place. Section III describes how colored rectangular zones are extracted from the document image. Section IV details the structural representation of both the whole document representation and the neighboring context of the searched information. Section V presents the experiments that have been conducted in order to evaluate our approach. Finally, section VI concludes the paper and draws some possible extensions to this work.

II. Context

Fig. 1 illustrates the main steps of the proposed approach. First, a GUI allows the user to surround the information he is interested in on a single sample document. From this acquisition, a structural representation is extracted to describe the neighboring context of the queried information. This representation constitutes the reading model. In production, the same structural representation is extracted on the whole image of the document to be processed. Thus, an error-tolerant subgraph isomorphism search is applied in order to find the subgraph in the whole document structural representation that matches the best with the neighboring context of the queried information. As a result, the instance of the queried information is located on the automatically processed document.

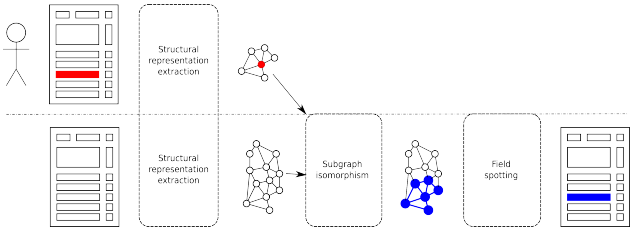

Figure 1. Overview of the framework

In a similar application context, Rusiñol *et al.* [1] propose an approach to overcome the variability in position and size of the queried information by locating the information w.r.t to its neighboring context. In this work, the neighboring context is modeled by a star graph whose center node represents the queried information and the branch nodes model words on the document that have been recognized by OCR. Then, spotting the queried information on a new document consists in (i) applying OCR, (ii) locating the word configuration that matches to the neighboring context of the queried information, (iii) deduce the expected position of the queried information. The importance of a peculiar word in the matching process of the neighboring context grows with its frequency in the context in learning documents. This approach gives performance that are related to the OCR results. Especially, it achieves good results when OCR operates in good conditions (clean binary images, typewritten words) but it may fail in other circumstances, in particular when it is faced to color documents. Some other works, as in [2] and [3], propose an attributed Relational graphs. Nevertheless, this kind of method is based on a semantic model annotated by the user unlike our system which is working from scratch and is only based on elements required by the user.

Nowadays, color documents constitute the vast majority of commercial and administrative forms (Fig. 2). Indeed, on this kind of documents, color lines, frames or rectangles are printed on the form background in order to guide the form filling as shown on Fig. 3.

Figure 2. Examples of documents

In the proposed approach, instead of being an indrance, we exploit color rectangles to model the neighboring context of the searched information considering that these are part of the

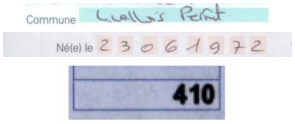

Figure 3. Examples of informative zones

fixed layout structure and can then be used as reference in the locating process. Section III and IV detail how the structural representation is constructed.

III. Informative zones retrieval

This section describes the first step in the construction of the proposed structural representation that consists in extracting colored rectangles. The proposed general processing procedure consists of three major steps: preprocessing, color quantization and selection of rectangular connected components.

A. Preprocessing

First, the preprocessing step aims at transforming the initial image in order to ease the following steps that lead to the rectangle extraction.

1) Edge preserving smoothing filter: Due to printing techniques of form backgrounds (four-color offset printing) and digitizing processes, the pixel values in digitized images do not correspond to the color that is perceived at a macrosopic scale. In order to give a better consistency to the pixel values and to ease the following color quantization process, we apply a filter on pixel values. For this purpose, we chose the Edge Preserving Smoothing Filter proposed by Nikolaou et al. in [4]. This adaptive mean filter adapts the smooting strength according to the color gradient. In that way, regions with low color gradient are smoothed (i.e. pixel values are brought closer to the mean value) whereas areas with high gradient are not smoothed and contour are kept sharp.

2) Inpainting: We also propose to apply an inpainting process to document images in order to ease the extraction of colored rectangles. This process aims at replacing the value of pixels belonging to text shape by the value of pixels surrounding these text elements. Indeed, the text elements, whether overprinted during the form filling or already present on the form background, may disturb the rectangle extraction as they reduce the number of pixels of rectangle shapes or even split that shapes in several components.

For that purpose, we use the smart binarization proposed by Gaceb *et al.* [5]. In the original image, the value of pixels classified as text are then replaced by the value of the nearest pixel classified as non text.

3) Color space: It is well known that the RGB color space is not always the most appropriate to detect requested information. Considering this point, before the color quatization step, the document image can be projected into alternative color spaces, namely $CIELab$ [6] or YIQ [7]. Once the input image is filtered, cleaned and/or projected into an other color space, we apply a color quantization in order to identify regions with homogeneous color.

B. Color quantization

The next step in the rectangle extraction is a color quantization process which may be conducted in any of the color spaces. This step aims at separating the preprocessed document image into several layers depending on the main perceived colors in the initial document. For this purpose, the $Kmeans$ algorithm clusters image pixels into K layers. We assume that the initial number of perceived colors is at least 3: one for the document background, at least one for the rectangles and finally, one for the text. However, the latter may have been eliminated by the inpainting step. So in our experiments, we investigate the influence of the parameter K on the performance.

C. Rectangle extraction

Once the image is transformed with a reduced number of colors, the last step consists in extracting rectangular shapes with the same color. To do this, in each color layer, we first extract the connected components. Then, the set of connected components is filtered so that only those with a rectangular shape are kept. According to [8], a region has a rectangular shape when the ratio of its surface with respect to the surface of its bounding box is greater than a threshold θ. The threshold θ is considered as the rectangularity criterion for a connected component. In our experiments, the threshold θ has been set to 0.9. One can notice that administrative and commercial documents are presented with a manhattan layout. As a consequence, in order to focus the search for only isothetic rectangles (*i.e.* with horizontal and vertical borders) the bounding box considered in the criterion computation is constrained to be also isothetic.

IV. Graph representation

Once color rectangles have been extracted using the strategy described in the previous section, the next step aims at building a graph representing their layout in the document. Let us recall that our objective with the graph representation is to locate informative zones through the description of their neighboring context. In this context, we choose to model the document as a visibility graph of rectangles.

A. Node description

As mentioned before, the graph nodes represent rectangles extracted by the first step. In order to complete the representation, some attributes are used to characterize the nodes. In the proposed approach, nodes are represented with two kinds of attributes :

1) Color attributes: Color information is mandatory to distinguish documents having similar structures but different colors. Thanks to an experimental study comparing the extraction performance in different color spaces, we have chosen to use the means of I and Q values (of the YIQ color space) in order to describe the color of a rectangle. From the RGB color space, I and Q pixel values are computed by :

$$I(p) = 0,596 \times R(p) - 0,275 \times G(p) - 0,321 \times B(p)$$

$$Q(p) = 0,212 \times R(p) - 0,528 \times G(p) - 0,311 \times B(p)$$

2) Geometric attributes: since rectangles have different sizes, their dimensions (width and height) is an important information to be taken into account in the matching process. In order to tolerate resolution difference, they are normalized by the size of the image.

Hence, each node in our graph is attributed with a numerical vector of 4 values : (I, Q, W, H).

B. Edge description

Nodes and their attributes are not sufficient to describe the layout of the document. Their topological relations are needed to circumscribe the information. In this context, we propose to go one step further than a simple neighbourhood graph through the use of pairwise topological properties based on visibility relations. Hence, two nodes are linked with an edge in the graph if they are considered as visible one from the other. Two rectangles are considered as visible if they share a sufficiently wide (resp. high) horizontal (resp. vertical) range of points that can be linked with a vertical (resp. horizontal) line without intersecting another rectangle.

Hence, let λ be a visibility rate, I_V the vertical visible intersection and I_H the horizontal one, an edge is created between two nodes $z_i(w_i, h_i)$ and $z_j(w_j, h_j)$ if

$$I_V \geq \lambda * min(w_i, w_j) \text{ or } I_H \geq \lambda * min(h_i, h_j) \quad (1)$$

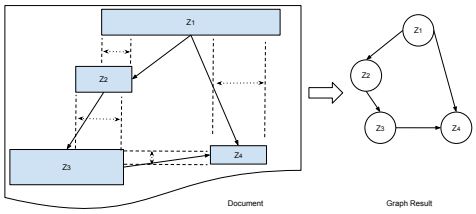

Figure 4. Visibility relations

Figure 4 illustrates the concept of a visibility relation on a toy problem, with $\lambda = 0.5$. In this figure, z_1 and z_2 are vertically visible because I_v is greater than half the width of z_2. The same configuration occurs for (z_2, z_3) and (z_1, z_4). On the contrary, z_1 and z_3 are not linked because there is no visibility between the rectangles. The visibility relation is attributed with the normalized horizontal and vertical distances between the rectangle, *i.e*, each edge has two attributes (V_D, H_D).

Using the strategy described in this section, we obtain an attributed graph $G = (\mathcal{V}, \mathcal{E}, \mu, \nu)$ where

- $\mathcal{V}$ is the (finite) set of vertices of G representing the rectangles,
- $\mathcal{E} \subseteq \mathcal{V} \times \mathcal{V}$ is the set of edges describing the visibility relations,
- $\mu : \mathcal{V} \rightarrow \mathcal{L}_\mathcal{V}$ is the vertex labeling function with $\mathcal{L}_\mathcal{V} = (I, Q, W, H)$, and
- $\nu : \mathcal{E} \rightarrow \mathcal{L}_\mathcal{E}$ is the edge labeling function with $\mathcal{L}_\mathcal{E} = (V_D, H_D)$.

Once graphs are built, we can apply a subgraph isomorphism in order to spot the interested zone. In the following section, we present our experiments and results.

V. Experiments and results

The experiments described in this section aim at assessing the relevance of the visibility graphs defined above in our framework of information spotting. In this context, we have designed an experimental protocol which simulates this framework. This protocol is described in the following subsection, before presenting and discussing the results.

A. Evaluation protocol

Our dataset is composed of 130 colored administrative and commercial document images divided into 8 classes (the distribution is given in the second line of table I). As explained in the introduction, the class of each document is considered as known in the scope of this paper. As a consequence, experiments are led independently in each class.

For each document of each class, 36 visibility graphs have been extracted. They have been built using different configurations of the zone retrieval step, the objective being to evaluate the impact of :

- the $EPSF$ pre-processing
- the $inpaint$ pre-processing
- the number K of layers
- the color space used for the clustering: RGB , YIQ or $CIELab$

In order to simulate the definition of the area of interest by the user, random graphs representing a query are built using 4 steps. First, the learning document is randomly chosen in the class. Then, a seed node representing the area of interest is randomly selected in this graph[1]. The query graph is thus completed with the nodes which are connected by an edge to the seed node. Finally, the query graph is completed by all the edges that link its nodes in the initial graph. In order to assess the generalization capacities of the approach, this process has been reproduced 50 times, leading to 50 query graphs for each class.

Using this query dataset, the subgraph isomorphism algorithm has been used to search for these 50 query graphs in all the documents of the class but the one used for building the query. This leads to 6100 queries for each configuration of the system.

For each query, the subgraph isomorphism algorithm outputs the set of node and edge matchings and the cost of the matching. The error-tolerant subgraph isomorphism used is an extension of the method presented in [9] which allows to match a node (resp. edge) from the pattern graph either to an actual node (resp. edge) in the target graph or to a created dummy node (resp. edge) if needed. In order to evaluate this result, it is necessary to check if the nodes that have been matched by the system really correspond to the same zones in the image. For that, we extract the list of substituted nodes S_{ij} and the list of created nodes C_i. We then compute the Jaccard distance $Jacc(Z_i, Z_j)$ between each pairs of nodes Z_i and Z_j in the S_{ij} list. For this computation, a ground truth is used to register both images. Then, we collect the correct matching list M_{ij} that validate the condition that $Jacc(Z_i, Z_j) > \alpha$. In the following, the threshold α is equal to 0.1. In the next step, we compute the matching score for a given query using equation 2.

$$match(q_i, t_j) = \frac{|M_{ij}|}{|S_{ij}| + |C_i|} \tag{2}$$

Then, in each class, we compute the performance of zone spotting using equation 3. We consider a subgraph is correctly matched when $match > \beta$. In our experience, we set $\beta = 0.5$, considering that a subgraph is satisfactorily located when more than 50% of nodes are correctly matched.

$$Perf = \frac{\sum_{i=1}^{50} \sum_{j=1}^{n-1} \mathbb{1}_{match(q_i,t_j)>\beta}}{50 * (n-1)} \tag{3}$$

where n is the number of documents in the class.

B. Results and discussion

Experimental results are presented in table I. This table presents the performance obtained for each class and for each processing configuration. A processing configuration is defined by setting faur parameters. The two first parameters concern the preprocessing step. They define whether or not EPSF and/or inpainting is applied. The two last parameters are relative to the color quatization step. The value of the parameter K is chosen in $\{2, 3, 4\}$ as justified in section III-B. Finally, the color space in which the $Kmeans$ algorithm operates is selected among $\{RGB, YIQ, CIELab\}$. This allows to test and compare 36 different processing configurations. The last column of I gives for each processing the performance observed on the whole dataset.

In order to help the reader, the best performance for each class is written in bold. We can observe that the best performance is always obtained when the inpainting preprocessing step is applied. We also noticed that the separation in two color layers leads in most of the case to the best result. From these observations, we emit the hypothesis that it is easier to split the pixels from the document image into two layers once those corresponding to the text layer have been removed. Finally, it is difficult to draw conclusions concerning the need for EPSF or the best color space. Indeed, depending on the class of the document to be processed the best performance is obtained with different combinations of these two parameters. This highlights the need to be able to automatically determine the configuration that works the best for a particular document.

Overall, the best processing configuration reaches a performance of 86.9% on the whole dataset. However, there exist a high diversity w.r.t the document class as informative zones are retrieved in 100% of cases for some classes, while only 72% are retrieved for other classes.

[1]For evaluation issues, the ROI is necessarily a node of the graph, i.e. a colored rectangle. However, the approach can be generalized to any rectagular zone of the image.

Table I. A SIMPLE EXAMPLE TABLE

Preprocessing		Color quantization		Classes								Base
EPSF	Inpaint	K	Color space	1(11)	2(31)	3(29)	4(14)	5(12)	6(11)	7(9)	8(13)	
yes	yes	2	RGB	58.85%	59.57%	75.44%	98.00%	99.33%	60.90%	69.33%	50.62%	69.67%
			YIQ	**83.45%**	89.35%	92.82%	**99.28%**	98.16%	44.72%	**92.66%**	58.29%	84.84%
			Lab	76.00%	82.00%	**93.44%**	98.57%	**100.00%**	45.27%	90.00%	45.98%	81.25%
		3	RGB	61.45%	65.54%	59.65%	90.29%	**100.00%**	78.54%	88.44%	42.92%	70.09%
			YIQ	79.81%	47.74%	72.13%	90.28%	**100.00%**	44.18%	43.77%	58.39%	65.77%
			Lab	68.54%	66.83%	84.96%	91.71%	97.16%	38.00%	56.22%	31.29%	69.70%
		4	RGB	70.90%	58.70%	66.62%	90.42%	**100.00%**	39.63%	61.11%	61.07%	67.46%
			YIQ	52.72%	55.87%	60.82%	84.85%	98.16%	37.63%	47.33%	49.53%	60.86%
			Lab	49.63%	32.38%	69.51%	88.71%	98.16%	45.81%	48.44%	38%	57.07%
	No	2	RGB	37.09%	25.03%	53.86%	62.64%	70.16%	21.09%	34.88%	30.15%	41.34%
			YIQ	65.45%	59.87%	82.41%	92.00%	92.66%	49.09%	83.55%	37.07%	70.21%
			Lab	76.00%	47.41%	78.27%	86.28%	93.00%	27.09%	60.22%	36.30%	63.16%
		3	RGB	74.00%	51.54%	43.93%	90.14%	74.67%	45.09%	65.11%	50.30%	58.13%
			YIQ	70.72%	62.96%	73.79%	87.71%	82.66%	40.36%	77.55%	51.07%	68.38%
			Lab	62.36%	62.32%	72.48%	87.14%	85.00%	34.90%	59.77%	41.07%	64.64%
		4	RGB	56.00%	61.93%	64.55%	87.85%	72.33%	38.72%	69.11%	24.92%	60.53%
			YIQ	43.63%	52.00%	58.75%	92.14%	77.50%	36.90%	60.66%	36.46%	57.24%
			Lab	56.36%	59.80%	64.55%	85.85%	75.16%	24.18%	65.33%	36.92%	59.84%
No	yes	2	RGB	59.09%	48.95%	59.37%	87.71%	99.16%	61.09%	62.88%	39.28%	61.94%
			YIQ	80.90%	**90.38%**	89.58%	99%	**100.00%**	**83.81%**	88.66%	55.21%	**86.90%**
			Lab	77.63%	74.96%	83.37%	90.57%	**100.00%**	52.36%	85.77%	47.82%	77.08%
		3	RGB	55.45%	65.03%	59.65%	87.85%	98.33%	83.63%	86.44%	29.09%	68.00%
			YIQ	74.70%	51.48%	85.37%	92.42%	**100.00%**	38.90%	36.44%	51.53%	67.78%
			Lab	50.00%	55.03%	79.93%	89.14%	95.50%	40.36%	50.88%	32.92%	63.83%
		4	RGB	79.45%	59.41%	70.96%	86.42%	95.16%	41.81%	48.44%	**72.00%**	68.93%
			YIQ	51.27%	41.35%	67.31%	93.42%	99.66%	42.36%	38.66%	43.84%	59.10%
			Lab	56.18%	34.77%	65.44%	90.28%	96.50%	34.90%	44.88%	31.07%	55.43%
	No	2	RGB	43.09%	22.83%	49.87%	55.31%	69.33%	31.81%	47.33%	23.84%	40.89%
			YIQ	63.63%	60.83%	78.48%	82.42%	89.83%	35.09%	69.55%	30.46%	65.35%
			Lab	58.00%	52.83%	64.82%	84.57%	84.66%	18.90%	64.66%	40.15%	58.95%
		3	RGB	63.63%	48.12%	59.51%	90.14%	78.16%	70.18%	65.55%	41.36%	61.53%
			YIQ	67.81%	72.45%	76.48%	85.85%	82.33%	40.18%	66.88%	46.46%	69.58%
			Lab	53.09%	56.83%	67.79%	85.00%	80.50%	46.36%	42.66%	46.00%	61.16%
		4	RGB	49.27%	59.74%	66.82%	85.00%	67.66%	52.54%	53.77%	24.30%	59.20%
			YIQ	56.00%	53.74%	62.27%	83.28%	79.83%	36.90%	59.11%	46.32%	59.49%
			Lab	52.90%	57.87%	50.55%	81.28%	74.83%	40.90%	59.33%	40.00%	56.66%

VI. CONCLUSION

In this paper, an approach for retrieving informative zones in administrative and commercial forms has been proposed. In order to tackle this problem, documents are modeled as graphs where nodes represent informative zones and edge is the visibility relations between zones. Having a subgraph as a query and a graph as a target, we apply an error-tolerant subgraph isomorphism algorithm in order to localize the information on other instances of document. Results show up that an efficient localization of ROI depends on the technique used to extract informative zones. In our future works, we propose to establish an evolutionary system to get the appropriate technique for each category of documents. Besides , further improvements can be done on the graph representation; we can use other information to expand the graph description or to replace colored frames.

REFERENCES

[1] M. Rusinol, T. Benkhelfallah, and V. P. d'Andecy, "Field extraction from administrative documents by incremental structural templates," in *Document Analysis and Recognition (ICDAR), 2013 12th International Conference on*. IEEE, 2013, pp. 1100–1104.

[2] C. A. Peanho, H. Stagni, and F. S. C. da Silva, "Semantic information extraction from images of complex documents," *Applied Intelligence*, vol. 37, no. 4, pp. 543–557, 2012.

[3] F. Cesarini, M. Gori, S. Marinai, and G. Soda, "Informys: A flexible invoice-like form-reader system," *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, vol. 20, no. 7, pp. 730–745, 1998.

[4] N. Nikolaou and N. Papamarkos, "Color reduction for complex document images," *International Journal of Imaging Systems and Technology*, vol. 19, no. 1, pp. 14–26, 2009.

[5] D. Gaceb, F. Lebourgeois, and J. Duong, "Adaptative smart-binarization method: For images of business documents," in *12th Intern. Conf. on Document Analysis and Recognition* IEEE, 2013, pp. 118–122.

[6] E. M. Sgarbi, W. A. Della Mura, N. M. J. Facon, and H. A. L. Ayala, "Restoration of old document images using different color spaces." in *Intern. Conf. on Computer Vision Theory and Applications, VISAPP*, Vol. 1, pp. 82–88, 2014.

[7] W.-Y. Chen and S.-Y. Chen, "Adaptive page segmentation for color technical journals' cover images," *Image and vision computing*, vol. 16, no. 12, pp. 855–877, 1998.

[8] P. L. Rosin, "Measuring rectangularity," *Machine Vision and Applications*, vol. 11, no. 4, pp. 191–196, 1999.

[9] P. Le Bodic, P. Héroux, S. Adam, and Y. Lecourtier, "An integer linear program for substitution-tolerant subgraph isomorphism and its use for symbol spotting in technical drawings," *Pattern Recognition*, vol. 45, no. 12, pp. 4214–4224, 2012.